B.Tech 3rd Year Roadmap PDF: Placement Prep

Third year is your placement preparation year. Finish DSA up to trees, graphs and dynamic programming basics, build one strong project you can explain in depth, revise core CS subjects and aptitude, and get a summer internship after 3rd year. Many colleges start placements early in 7th semester, so be ready by then.

Key takeaways

- Treat 3rd year as preparation: by its end you should be able to sit any placement test.
- One strong, deployed project beats five copied ones.
- The summer internship after 3rd year is often the most valuable one, and some lead to pre-placement offers.
- Practise aptitude and communication too. Many first rounds are aptitude tests.
- Decide early if you also want GATE, higher studies or a core job, so you can plan time.

Why 3rd year is the turning point

In many colleges, campus placements begin in the 7th semester, which is just after 3rd year. Some companies visit even earlier for internships. That means your 3rd year effort shows up directly in your first offers. It is also the year when subjects get heavier, so planning matters.

The aim is simple: by the end of 6th semester, you should be able to take an online coding test, an aptitude test and a technical interview without panic.

The five goals for year 3

1. **Complete DSA:** trees, binary search trees, heaps, graphs (BFS, DFS, shortest path), greedy and dynamic programming basics.
2. **One strong project** in your chosen track, deployed and explained well.
3. **Core CS revision:** DBMS, OS, computer networks, OOP and basic system design ideas.
4. **Aptitude and communication:** quantitative, logical reasoning, verbal and group discussion practice.
5. **A summer internship** after 3rd year.

Semester 5: deepen skills

Months 1 to 2

- DSA: trees, binary search trees, heaps. Solve medium problems, not only easy ones.

- Track: go deeper. For web, add a backend, authentication and a database. For data, add SQL depth, dashboards and basic statistics. For testing, add a framework and API testing.
- Start aptitude practice: 20 to 30 minutes, three times a week.

Months 3 to 4

- DSA: graphs and greedy algorithms.
- Start your strong project. Choose a real problem: a tool for your college, a small business or a club.
- Update your resume and LinkedIn profile.
- Apply for summer internships. Many companies open applications months ahead.

Exam month

- Keep CGPA steady. A sudden drop in 5th semester is visible to recruiters.

Semester 6: test yourself

Months 1 to 2

- DSA: dynamic programming basics and revision of all topics.
- Finish and deploy your project. Write a README with screenshots, setup steps and what you learned.
- Take timed mock tests: coding, aptitude and MCQs on core subjects.

Months 3 to 4

- Practise mock interviews with friends or seniors. Explain your project in 2 minutes and in 10 minutes.
- Revise DBMS, OS and networks using short notes.
- Prepare HR answers honestly: strengths, weaknesses, why this company, and how you handled a problem.

Summer after 3rd year

- Do your internship well: ask questions, take notes and finish tasks. Some companies make pre-placement offers to good interns, but it is never certain.
- Continue a light DSA and aptitude routine during the internship.

Sample weekly plan (semester time)

Day	DSA (60 min)	Project (60 min)	Other (30 min)
Monday	2 medium problems	Feature work	Aptitude
Tuesday	1 topic revision	Feature work	Core subject notes
Wednesday	2 problems	Testing and bug fixes	Aptitude
Thursday	2 problems	Feature work	Communication practice
Friday	Contest or timed test	Documentation	Core subject notes

Day	DSA (60 min)	Project (60 min)	Other (30 min)
Saturday	Mock test, 2 hours	Project, 2 hours	Mock interview
Sunday	Review mistakes	Rest	Plan next week

Should you also prepare for GATE or higher studies?

Decide by the middle of 3rd year. GATE preparation takes many months alongside college, so it needs a clear plan. If you are interested in M.Tech, PSU jobs or research, read our GATE strategy guides. If you want MS abroad, start shortlisting universities and test dates in 3rd year. It is fine to keep placements as a backup, but do not try to do everything at full speed.

What makes a project "strong"

A strong project is not about using the most tools. Recruiters and interviewers tend to look for:

- **A real problem.** "A leave request system for our department" is better than "an e-commerce clone".
- **Completeness.** Users can sign up, do the main task and see results. It handles wrong input without crashing.
- **Deployment.** It runs on a public link, or there is a short demo video if it cannot be hosted.
- **Your understanding.** You can explain why you chose the database, how login works and what you would change for 10 times more users.
- **Clean code and history.** Meaningful commit messages, a README and no secret keys in the repository.

Preparing for the three common rounds

Most campus processes include some mix of these rounds. The exact format varies by company, so read each notice.

1. **Online assessment:** aptitude, MCQs on core subjects and coding questions under time limits. Practise in timed conditions on a laptop, not only on paper.
2. **Technical interview:** DSA questions, your project, core subjects and sometimes SQL queries. Think aloud and explain your approach before you code.
3. **HR interview:** background, strengths, willingness to relocate, and questions about the company. Be honest and specific. Research the company the day before.

Keep a mistake log after every mock test: the question, why you got it wrong and the correct idea. Revise this log weekly. It is the fastest way to improve.

What to avoid in 3rd year

- Starting DSA from zero in 6th semester. It is possible, but stressful.
- Listing technologies on your resume that you cannot answer questions on.

- Skipping aptitude practice because you are "good at coding". Many students are filtered in the aptitude round.
- Ignoring backlogs. Many companies ask for no active backlogs at the time of placement.
- Learning only from short videos. Read official documentation for the tools you use; interviewers notice the difference.
- Skipping sleep before tests. A rested mind solves problems faster than a tired one that revised all night.
- Waiting for the placement cell to tell you what to do. Read past placement notices from your college, talk to seniors placed last year and plan from what they faced.

End of year 3 checklist

- [ ] DSA covered up to graphs and DP basics, with regular practice
- [ ] One strong deployed project with a clear README
- [ ] Summer internship done or in progress
- [ ] Core CS subjects revised with short notes
- [ ] Aptitude mock tests taken under time limits
- [ ] Resume reviewed by at least two people
- [ ] Decision made on placements, GATE, higher studies or a mix

Uncodemy's Full Stack Development course includes project work and mock interviews, and Uncodemy provides Dedicated Placement Support to learners.

Frequently asked questions

What should I do in B.Tech 3rd year for placements?

Finish DSA up to graphs and DP basics, build one strong deployed project, revise core subjects, practise aptitude and do a summer internship. Be test-ready by the end of 6th semester.

Is 3rd year too late to start coding?

It is late but not too late. Start with one language and DSA basics right away, keep a daily routine, and focus on one project rather than many.

How important is the 3rd year summer internship?

It is often the most useful internship because it is close to placements. It gives real work to talk about, and some companies offer pre-placement offers to good interns.

Should I prepare for GATE and placements together?

Many students do, but it needs a clear timetable. Decide your main goal by mid 3rd year and give it most of your time.

How many projects do I need by the end of 3rd year?

One strong project you can explain in depth, plus one or two smaller ones, is usually enough. Quality and understanding matter more than count.

Do I need to learn system design in 3rd year?

Only the basics: how a web app, database and server work together, caching and scaling ideas. Deep system design is asked more for experienced roles.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- CSES Problem Set (free DSA practice)
- NPTEL: free IIT and IISc courses
- AICTE Internship Portal
- Pro Git book (free, official Git documentation)

About this PDF

Prepared by the Uncodemy career team from the public sources listed. We review it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550